

REMARKS

In the Office Action, the Examiner rejected Claims 1-24, which are all of the pending claims, under 35 U.S.C. 103 as being unpatentable over U.S. Patent 6,219,690 (Slingwine); and Claims 10, 11 and 13-17 were further rejected under 35 U.S.C. 101 as directed to non-statutory subject matter. Claims 10-17 were also rejected under 35 U.S.C. 112, first paragraph, as not complying with the written description requirement; and under 35 U.S.C. 112, second paragraph, as being indefinite. In addition, Claims 6, 14 and 21 were rejected under the doctrine of obviousness-type double patenting as being unpatentable over claim 7 of copending application no. 11/227,761, and the Examiner objected to language used in the specification.

Independent Claims 1, 10 and 18 are being amended to better define the subject matters of thee claims. Claims 10 is being amended to address the rejection of Claims 10, 11 and 13-17 under 35 U.S.C. 101; and Claims 10 and 17 are being amended to address the rejections of Claims 10-17 under 35 U.S.C. 112, first and second paragraphs. Claims 13-17 are being amended to keep the language of these claims consistent with the language of Claim 10. The specification is being amended to change the language to which the Examiner objected.

Also, Applicants respectfully request that the double patenting rejection of Claims 1, 10 and 18 be held in abeyance pending an indication that the claims patentably distinguish over the prior art.

In objecting to the specification, the Examiner, in particular, objected to the term “source code” in the Summary section of the application. After reviewing the application, the Summary is being amended, consistent with the Detailed Description section of the application, to refer to “first source code component” to “first code component,” and to change “new source code component” to “new code component.” It is believed that this change eliminates the misuse of

terminology in the Summary section, and the Examiner is, accordingly, asked to reconsider and to withdraw the objection to the specification.

With respect to the rejection of Claims 10, 11 and 13-17 under 35 U.S.C. 101, the Examiner, in the Office Action, argued that these claims are directed to non-statutory subject matter because the claims fail to provide hardware support to carry out the software functionality described in the claims.

Claims 11 and 13-17 are all dependent from Claim 10, which, as amended herein, is directed to a system for swapping code components in a computer system. Claim 10 is being amended to indicate that this swapping system includes one or more processor units that are configured to perform the functions positively set forth in the body of the claim. These features include instantiating an instance of a new source code component to replace a first code component, and establishing a quiescent state for the first code component. Thus, as now amended, Claim 10 positively sets forth physical hardware – the one or more processing units – to carry out the functionality described in the claim.

In view of the foregoing, all of Claims 10, 11 and 13-17 include hardware to carry out the claimed functions, and all of these claims are directed to statutory subject matter within the meaning of 35 U.S.C. 101. The Examiner is thus asked to reconsider and to withdraw the rejection of Claims 10, 11 and 13-17 under 35 U.S.C. 101.

After carefully reviewing the Office Action and the claims, it appears that the rejections of Claims 10-17 under 35 U.S.C. 112, both first and second paragraphs, is due to use in the claims of the term “source code.” The Examiner indicated, for instance, that it is not clear how source code is swapped, and also that in the claims, that there is insufficient antecedent basis in the claims for both “first source code component” and “first code component.”

Claims 10-17 have been carefully reviewed to remove references to “source code” and to describe the code components more consistently. For example, in the preamble of the claim, “source code component” is being changed to “code component.” The preamble of Claim 10 also sets forth “at least one code component,” and the body of the claim is being amended to change “first code component” to “the one code component,” which refers to the at least one code component set forth in the preamble of the claim.

Moreover, Applicants respectfully submit that removing the reference in Claim 10 to “source code” obviates the rejection of Claims 10-17 as not being enabled by the specification. In particular, the specification fully enables swapping code components, as now set forth in Claims 10-17. This procedure is discussed in detail on pages 4-15 of the application.

Applicants’ Attorneys have carefully reviewed all of Claims 10-17, and these claims, as now amended, are clear and definite and also are fully enabled by the specification. Consequently, the Examiner is requested to reconsider and to withdraw the rejections of Claims 10-17 under 35 U.S.C. 112, first and second paragraphs.

Furthermore, all of Claims 1-24 patentably distinguish over the prior art; and the Examiner is also asked to reconsider and to withdraw the rejection of Claims 1-24 under 35 U.S.C. 103, and to allow these claims.

Generally, Claims 1-24 patentably distinguish over the prior art because the prior art does not disclose or render obvious the way in which code is “hot swapped” a computer system – that is swapped while requests are made to the code and those requests are performed- as described in the independent Claims 1, 10 and 18. More specifically, the prior art does not disclose or render obvious using a mediator object for intercepting calls requesting the code component to perform

functions, and using the mediator object to perform the requested functions, as described in those independent claims.

In order to best understand this feature and its significance, it may be helpful to review briefly this invention and the prior art.

As discussed in the present application, operating systems are large and complex, and must satisfy a number of difficult requirements. For instance, these systems are expected to run across a varied set of hardware platforms, are expected to stay up for long periods of time, and are expected to stay up to date with the latest fixes. Further, these operating systems serve an increasingly divergent set of application workloads.

In the past, operating systems programmers have used several approaches to attempt to achieve some of these goals, and there is a large body of prior work focusing on the downloading and dynamic binding of new components. There has, though been less work on swapping of transparent scalable components in an active system. In particular, the prior art has not been able to provide a generic hot-swapping capability for operating systems that allows them to activate new code without stopping the service, while also performing effectively under a varying set of workloads and while ensuring continuous availability.

The present invention effectively addresses this need.

Generally, this is done by allowing new code, which has been downloaded to fix or upgrade a particular service, to be enabled and activated in a computer operating system without having to bring down the system or the service.

More specifically, in one embodiment, the invention provides a method of replacing a first code component in a computer system with a new code component while the operating

system of the computer remains active and while that operating system provides continual availability to hardware resources by applications operational in the computer system.

In this method, an instance of a new code component to replace the first code component is instantiated. A mediator object is instantiated, and a swapping operation is initiated. For the duration of the swapping operation, the mediator object is used to intercept calls from the applications requesting the first code component to perform defined functions, and also to perform the requested functions. A quiescent state is established for the first code component, state is transferred from the first code component to the new code component, and the new code component is then swapped for the first code component. This swapping includes identifying references to the first code component, and replacing the identified references to that first code component with references to the new code component.

The prior art does not disclose or render obvious using a mediator object as described above – that is, to intercept calls from the applications requesting the first code component to perform defined functions, and also to perform the requested functions.

For instance, Slingwine, et al, which is the only reference relied on by the Examiner to reject the claims, describes a procedure for updating data. This procedure allows concurrent reading and updating data while maintaining data coherency.

In the Slingwine, et al. procedure, a mutual-exclusion mechanism tracks a thread execution history to determine safe times for processing a current generation of data updates while a next generation of data updates is concurrently being saved. A summary of thread activity tracks which threads have passed through a quiescent state after the current generation of updates was started. When the last thread related to the current generation passes through a quiescent state, the summary of thread activity signals a callback processor that it is safe to end

the current generation of updates. This callback processor then processes and erases all updates in the current generation, and the next generation of updates then becomes the current generation of updates.

As Applicants noted previously, there is a very important general difference between the present invention and the procedure disclosed in Slingwine, et al – the present invention is directed to swapping code, while Slingwine, et al. is directed to updating data.

This general difference is reflected in a number of more specific differences between the present invention and the method and system disclosed in Slingwine, et al.

One very important difference is that Slingwine, et al. does not show the use of Applicants' mediator object, as described above. Moreover, there is no reason to do this in Slingwine, et al. because in the procedure disclosed therein, the old data is not called upon to perform functions – that old data is being replaced with the new data. With the present invention, in contrast, when code is replacing code, it is critical that the functions performed by the old code continue to be performed until the system is ready to have the new code perform these functions.

Independent Claims 1, 10 and 18 are being amended to emphasize the above-discussed feature of the present invention. In particular, each of these claims is being amended to describe the feature that for the duration of the swapping operation, the mediator object is used to intercept calls from the applications requesting the first, or the old, code component to perform defined functions, and also to perform the requested functions.

The other references of record have been reviewed, and these other references, whether considered individually or in combination, also do not show or render obvious this aspect of the present invention.

In view of the above-discussed differences between Claims 1, 10 and 18 and the prior art, and because of the advantages associated with these differences, Claims 1, 10 and 18 patentably distinguish over the prior art and are allowable. Claims 2-9 are dependent from Claim 1 and are allowable therewith. Also, Claims 11-17 are dependent from Claim 10 and are allowable therewith; and Claims 19-24 are dependent from, and are allowable with, Claim 18. The Examiner is thus asked to reconsider and to withdraw the rejection of Claims 1-24 under 35 U.S.C. 103 and to allow these claims.

For the reasons advanced above, the Examiner is asked to reconsider and to withdraw the objection to the specification, the rejections of Claims 10, 11 and 13-17 under 35 U.S.C. 101, and the rejections of Claims 10-17 under 35 U.S.C. 112, first and second paragraphs. The Examiner is also asked to reconsider and to withdraw the rejection of Claims 1-24 under 35 U.S.C. 103, and to allow these claims. If the Examiner believes that a telephone conference with Applicants' Attorneys would be advantageous to the disposition of this case, the Examiner is asked to telephone the undersigned.

Respectfully submitted,

John S. Sensny
John S. Sensny
Registration No. 28,757
Attorney for Applicants

SCULLY, SCOTT, MURPHY & PRESSER, P.C.
400 Garden City Plaza – Suite 300
Garden City, New York 11530
(516) 742-4343

JSS:jy